DOCUMENT RESUME

ED 056 533 EM 009 394

TITLE	Project Solo; Newsletter Number Thirteen.
INSTITUTION	Pittsburgh Univ., Pa. Dept. of Computer Science.
SPONS AGENCY	National Science Foundation, Washington, D.C.
PUB DATE	Mar 71
NOTE	19p.; See also ED 053 566
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	Computer Assisted Instruction; *Computer Graphics; *Computer Programs; *Computer Science Education; German; Programing; *Student Developed Materials; *Vocabulary Development
IDENTIFIERS	*Project Solo

ABSTRACT

A computer-assisted drill program designed to give German vocabulary drill is presented. It was written by a high school student and is an example of a file oriented program in which the main master program remains fixed while the vocabulary words to be drilled can be changed at any time either by the teacher or by the student. Some polar coordinate plotter routines for use on the Hewlett-Packard plotter are also presented. (JY)

ED056533

PROJECT SOLO

AN EXPERIMENT IN REGIONAL COMPUTING
FOR SECONDARY SCHOOL SYSTEMS*

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

University of Pittsburgh • Department of Computer Science • Pittsburgh, Pennsylvania 15213

Newsletter No. 13 March 1971

File Oriented CAI auf Deutsch

The term CAI (Computer Assisted Instruction) originally had a fairly restricted meaning. It implied that a "master program" written by a professional author was run by a student who was then led through drill and practice type work. The Project SOLO philosophy is to use such programs where appropriate, but to always make the master program available to the student for inspection, modification, and often considerable improvement.

We are enclosing an example of such a drill program written by a high school freshman at Fox Chapel, Mark Pilant. Mark designed this program entirely on his own initiative after hearing a teacher of German ask whether vocabulary drill might be possible through time-sharing. Miss Kavanagh provided terminal time and some programming guidance.

The professional aspect of Mark's program is the way in which it uses files. While the main master program remains fixed, the vocabulary words to be drilled can be changed at any time, either by the teacher, or preferably by the student, who is quite literally in full control of what he wants to drill at any given time. There are also separate files for German reinforcement messages which can be modified by students or teachers to provide some real surprises. This is nothing short of a stroke of psychological genius (we've been running it in the project office just to learn different ways of saying 'great' and 'lousy' auf Deutsch).

More Plotter Routines

We are also enclosing some polar coordinate plots done on the H-P plotter. Mr. Valenti & Mr. Gillespie use these with their trigonometry classes as a laboratory demonstration device. Two other plotter routines developed by Frank Wimberly are also enclosed. The first draws a large number of tangents to a curve (sine function in his example). The second uses the plotter to simulate a bouncing ball in a three-walled court. A billiard table plotter simulation and a pantograph analysis program will be coming in a later newsletter.

---

*Supported in part by NSF grant GJ 1077

EM 009 394

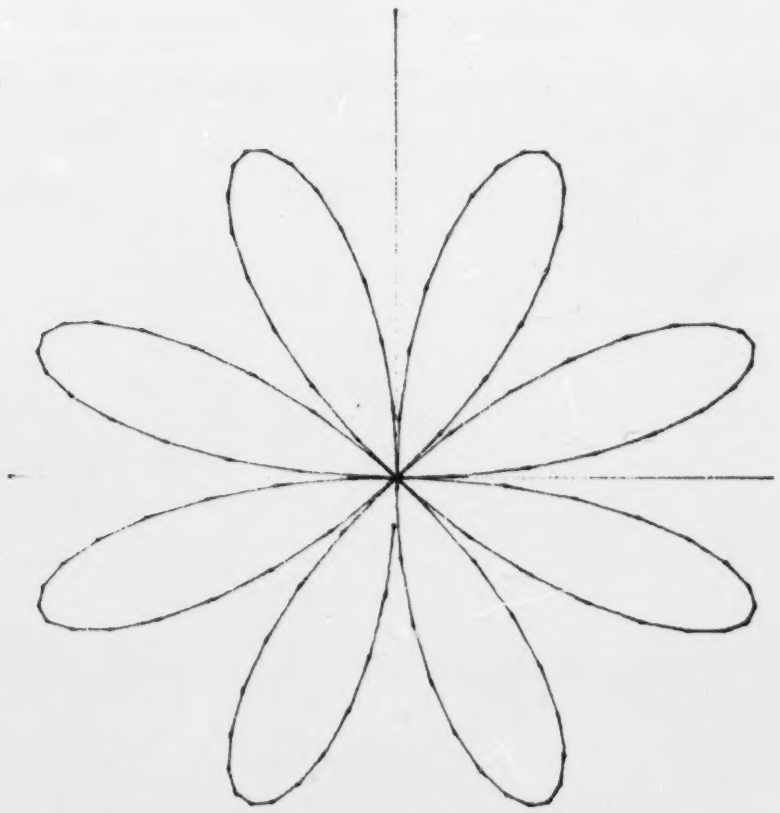
LISTING OF THE PROGRAM /SIN-PETAL/

```
1 PR."PLTL"
2 PR. 500:7000
3 PR. 4500:7000
4 PR."PLTP"
5 PR. 2500:9000
6 PR. "PLTL"
7 PR. 2500:5000 PR."PLTT"
8 PR."PLTL"
9 FØR I=0 TØ 2*PI BY .05
10 R=SIN(2*I)
11 Y=2000*R*SIN(I) LET X=2000*R*CØS(I)
12 PR. INT(.755*X)+2500:INT(Y)+7000
13 NEXT I
14 PR. "PLTP"
15 PR. 5000:7000
16 PR. "PLTL"
17 PR. 9000:7000
18 PR. 7000:9000:"↑"
19 PR. "PLTL"
20 PR. 7000:5000 PR."PLTT"
25 PR."PLTL"
26 FØR I=0 TØ 2*PI BY .05
27 R=SIN(4*I)
28 X=2000*R*CØS(I) LET Y=2000*R*SIN(I)
29 PR. INT(.755*X)+7000:INT(Y)+7000
30 NEXT I
32 PR. "PLTP"
33 PR. 500:2500
34 PR."PLTL"
35 PR. 4500:2500
36 PR. 2500:4500:"↑"
37 PR. "PLTL"
38 PR. 2500:500 PR."PLTT"
39 PR."PLTL"
40 FØR I=0 TØ PI BY .05
41 R=SIN(3*I)
42 X=2000*R*CØS(I) LET Y=2000*R*SIN(I)
43 PR. INT(.755*X)+2500:INT(Y)+2500
44 NEXT I
45 PR. "PLTP"
46 PR. 7000:4500
47 PR. "PLTL"
48 PR. 7000:500
49 PR. 5000:2500:"↑"
50 PR."PLTL"
51 PR. 9000:2500 PR."PLTT"
52 PR."PLTL"
53 FØR I=0 TØ PI BY .025
54 R=SIN(5*I)
55 X=2000*R*CØS(I) LET Y=2000*R*SIN(I)
56 PR. INT(.755*X)+7000:INT(Y)+2500
57 NEXT I
58 PR."PLTT"
60 END
```

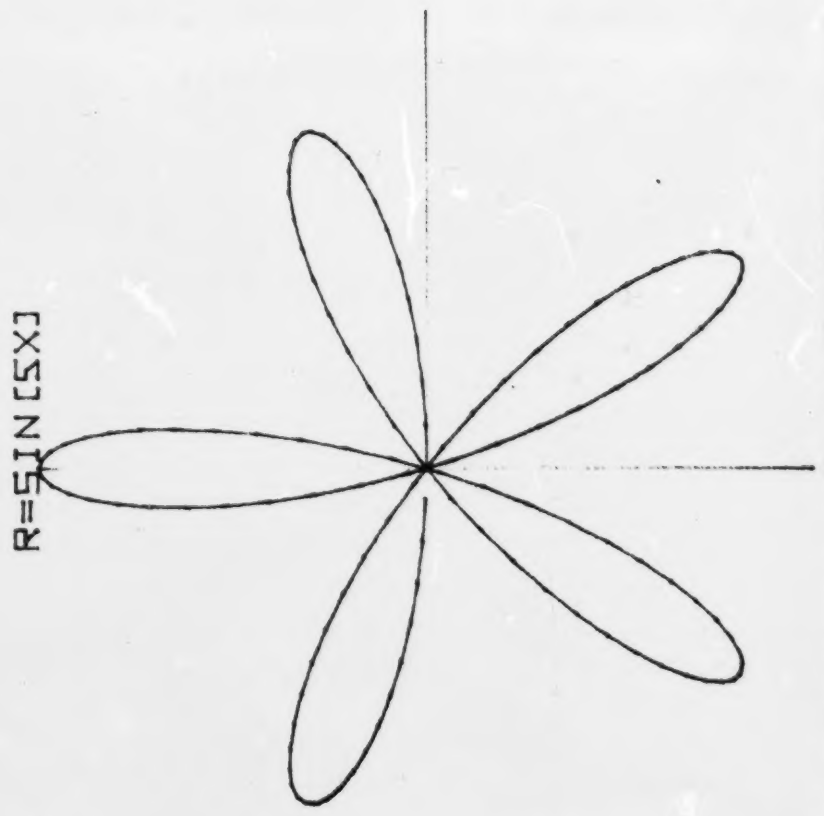
Note: Multiplying factor for X should be .66667 if you wish perfect symmetry on a 10 x 15" field.

THE PLOTTED OUTPUT FROM A RUN OF /SIN-PETAL/

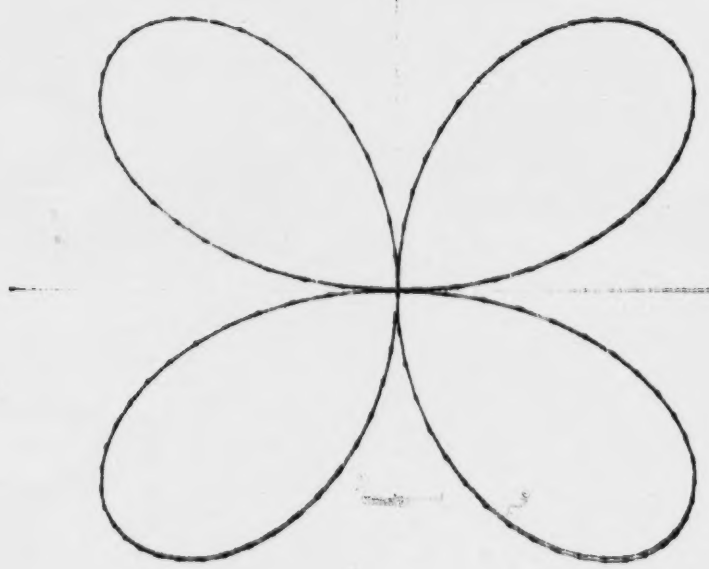
R=SIN[4X]



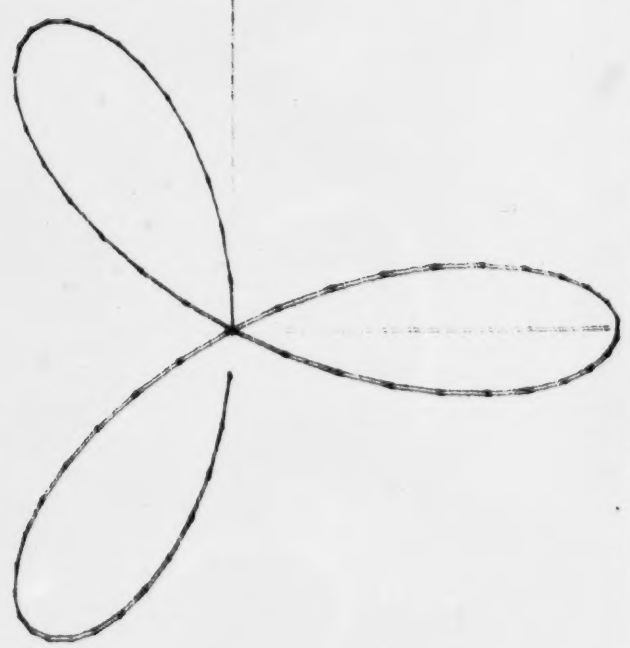
R=SIN[5X]



R=SIN[2X]



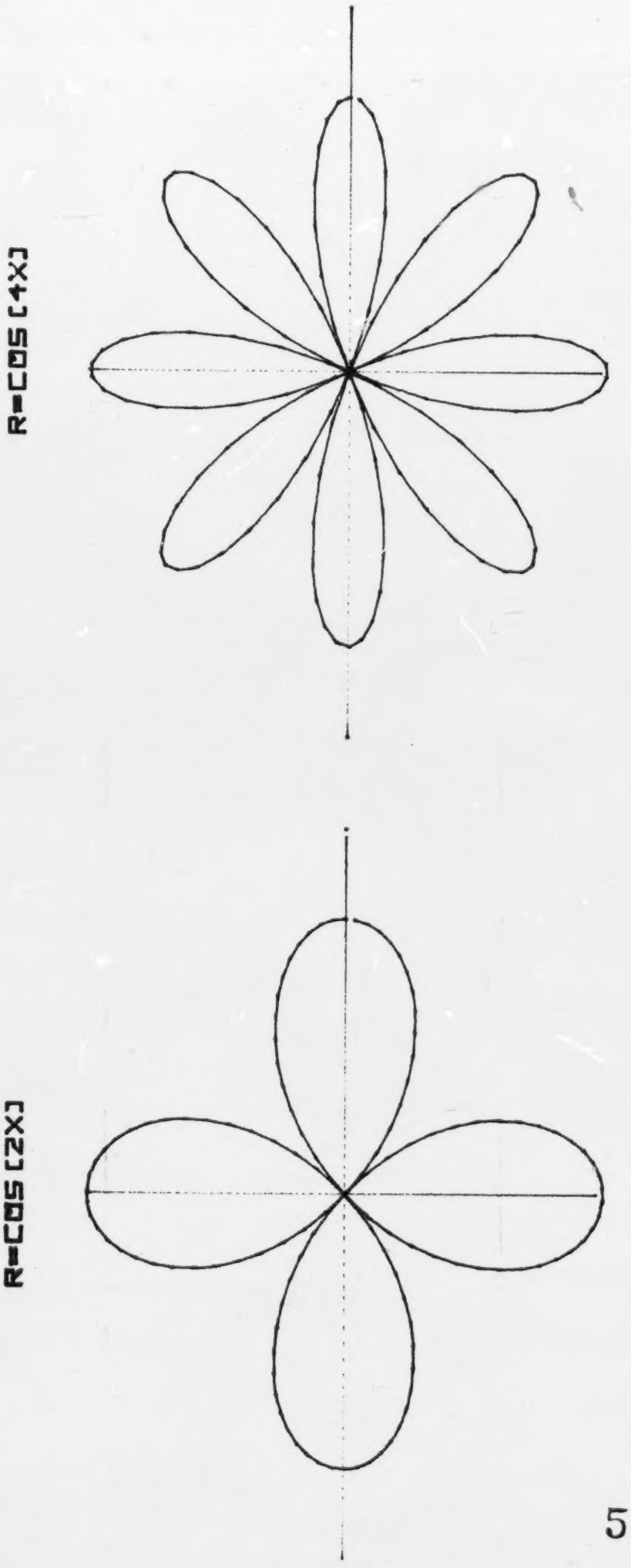
R=SIN[3X]



LISTiNG OF THE PROGRAM /COS-PETAL/

```
1 PR."PLTL"
2 PR. 500:7000
3 PR. 4500:7000
4 PR."PLTP"
5 PR. 2500:9000
6 PR. "PLTL"
7 PR. 2500:5000 PR."PLTT"
8 PR."PLTL"
9 FØR I=0 TØ 2*PI BY .05
10 R=CØS(2*I)
11 Y=2000*R*SIN(I) LET X=2000*R*CØS(I)
12 PR. INT(.755*X)+2500:INT(Y)+7000
13 NEXT I
14 PR. "PLTP"
15 PR. 5000:7000
16 PR. "PLTL"
17 PR. 9000:7000
18 PR. 7000:9000:"↑"
19 PR. "PLTL"
20 PR. 7000:5000 PR."PLTT"
25 PR."PLTL"
26 FØR I=0 TØ 2*PI BY .05
27 R=CØS(4*I)
28 X=2000*R*CØS(I) LET Y=2000*R*SIN(I)
29 PR. INT(.755*X)+7000:INT(Y)+7000
30 NEXT I
32 PR. "PLTP"
33 PR. 500:2500
34 PR."PLTL"
35 PR. 4500:2500
36 PR. 2500:4500:"↑"
37 PR. "PLTL"
38 PR. 2500:500 PR."PLTT"
39 PR."PLTL"
40 FØR I=0 TØ PI BY .05
41 R=CØS(3*I)
42 X=2000*R*CØS(I) LET Y=2000*R*SIN(I)
43 PR. INT(.755*X)+2500:INT(Y)+2500
44 NEXT I
45 PR. "PLTP"
46 PR. 7000:4500
47 PR. "PLTL"
48 PR. 7000:500
49 PR. 5000:2500:"↑"
50 PR."PLTL"
51 PR. 9000:2500 PR."PLTT"
52 PR."PLTL"
53 FØR I=0 TØ PI BY .025
54 R=CØS(5*I)
55 X=2000*R*CØS(I) LET Y=2000*R*SIN(I)
56 PR. INT(.755*X)+7000:INT(Y)+2500
57 NEXT I
58 PR."PLTT"
60 END
```

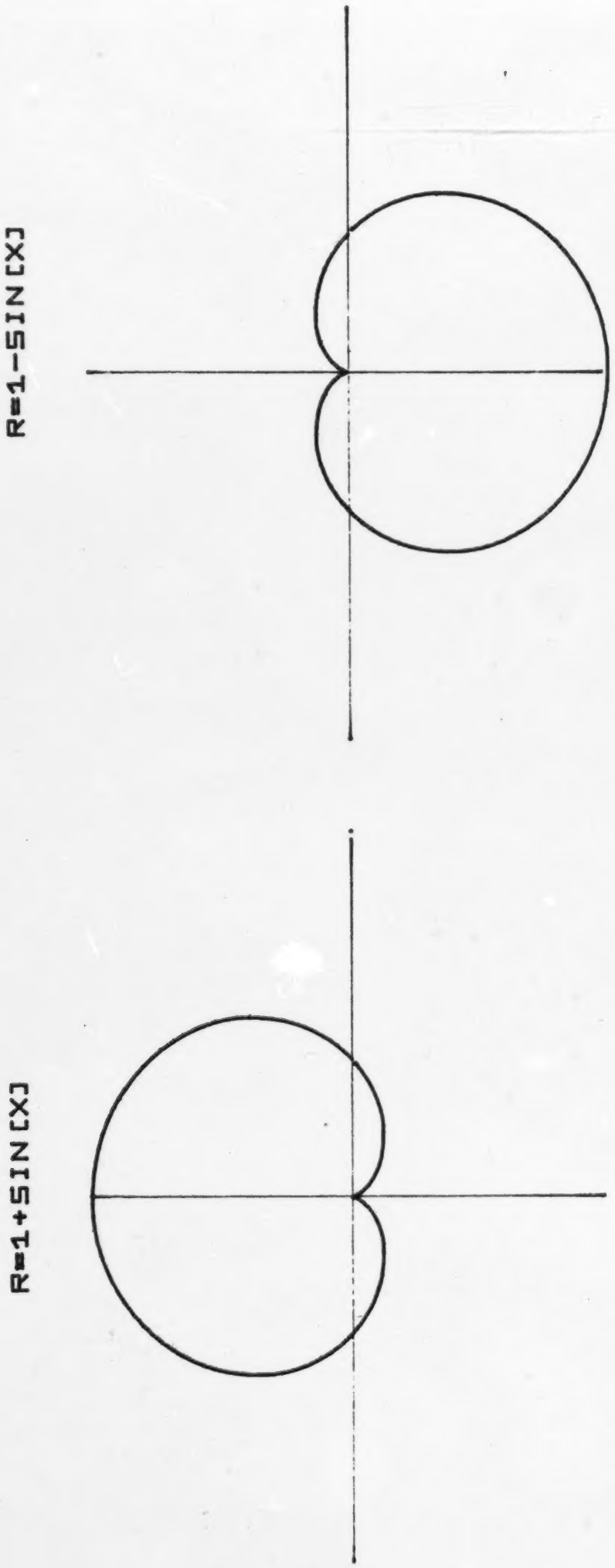
THE PLOTTED OUTPUT FROM A RUN OF /COS-PETAL/



LISTING OF THE PROGRAM /CARD/

```
1 PR."PLTL"
2 PR. 500:7000
3 PR. 4500:7000
4 PR."PLTP"
5 PR. 2500:9000
6 PR. "PLTL"
7 PR. 2500:5000 PR."PLTT"
8 PR."PLTL"
9 FØR I=0 TØ 2*PI BY .05
10 R=1+SIN(I)
11 X=1000*R*CØS(I),Y=1000*R*SIN(I)
12 PR. INT(.755*X)+2500:INT(Y)+7000
13 NEXT I
14 PR. "PLTP"
15 PR. 5000:7000
16 PR. "PLTL"
17 PR. 9000:7000
18 PR. 7000:9000:"↑"
19 PR. "PLTL"
20 PR. 7000:5000 PR."PLTT"
25 PR."PLTL"
26 FØR I=0 TØ 2*PI BY .05
27 R=1-SIN(I)
28 X=1000*R*CØS(I),Y=1000*R*SIN(I)
29 PR. INT(.755*X)+7000:INT(Y)+7000
30 NEXT I
32 PR. "PLTP"
33 PR. 500:2500
34 PR."PLTL"
35 PR. 4500:2500
36 PR. 2500:4500:"↑"
37 PR. "PLTL"
38 PR. 2500:500 PR."PLTT"
39 PR."PLTL"
40 FØR I=0 TØ 6.28 BY .05
41 R=1+CØS(I)
42 X=1000*R*CØS(I),Y=1000*R*SIN(I)
43 PR. INT(.755*X)+2500:INT(Y)+2500
44 NEXT I
45 PR. "PLTP"
46 PR. 7000:4500
47 PR. "PLTL"
48 PR. 7000:500
49 PR. 5000:2500:"↑"
50 PR."PLTL"
51 PR. 9000:2500 PR."PLTT"
52 PR."PLTL"
53 FØR I=0 TØ 6.28 BY .05
54 R=1-CØS(I)
55 X=1000*R*CØS(I),Y=1000*R*SIN(I)
56 PR. INT(.755*X)+7000:INT(Y)+2500
57 NEXT I
58 PR."PLTT"
60 END
```

THE PLOTTED OUTPUT FROM A RUN OF /CARD/



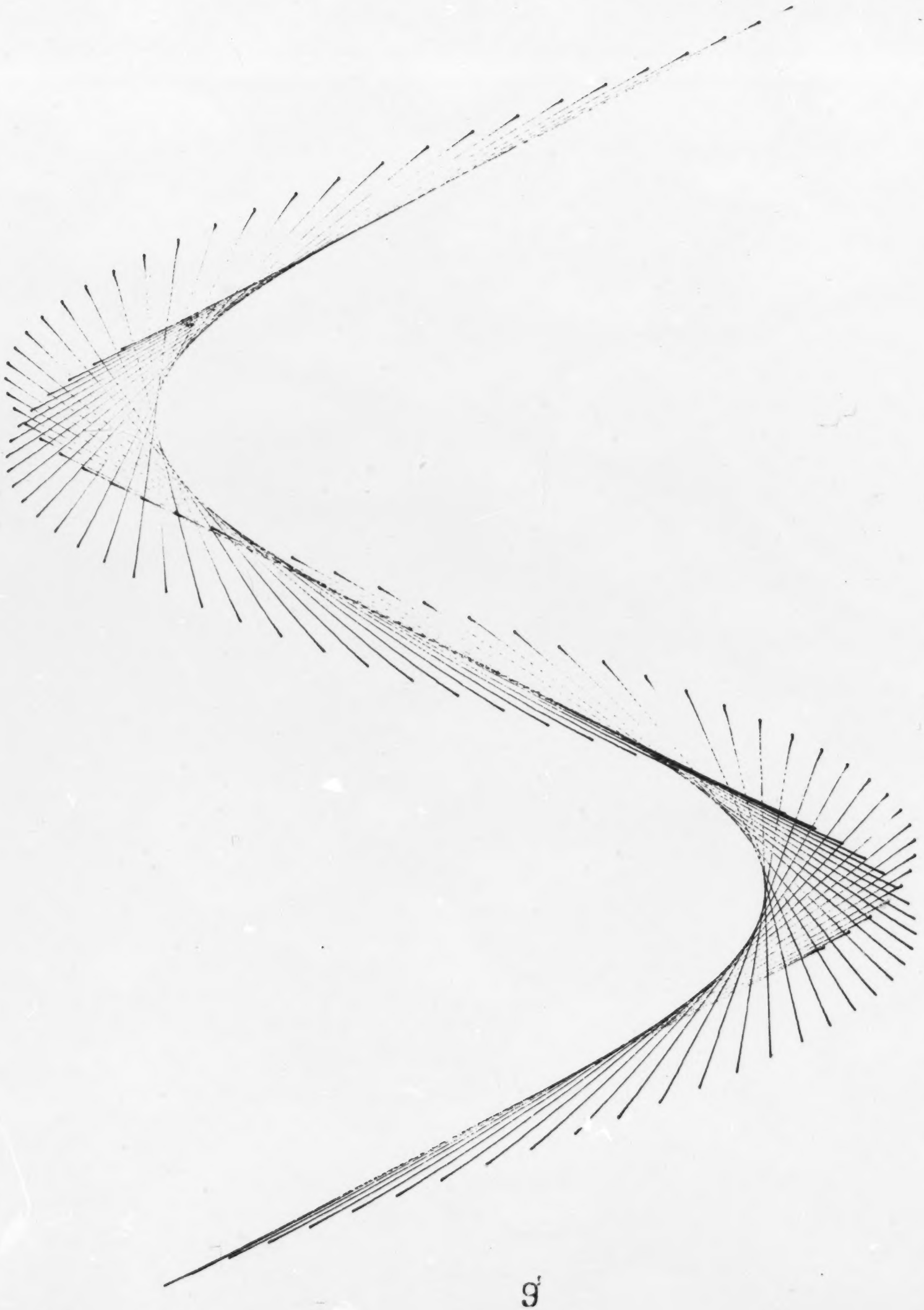
LISTING OF /SINTAN/

```
 FEB  8 11:57   /SINTAN/

10 PRINT "PLTL"
20 FØR X=-3.1416 TØ 3.1416 STEP .1
30 X1=5000 +900*X
40 Y1=2500*SIN(X)+5000
50 X2=X1-1000
60 Y2=Y1-CØS(X)*2777
7
 0 PRINT INT(X2); INT(Y2): ";"
80 X3=X1+1000
90 Y3=Y1+CØS(X)*2777
100 PRINT INT(X3); INT(Y3)
110 NEXT X
120 PRINT "PLTT"
```

EXPLANATION OF THE PROGRAM /SINTAN/

STATEMENT	
10	Start plotting in line mode.
20	Do calculations for x=-π to π in steps of .1.
30	Scale and shift value of x. for range of plotter (0 to 9999).
40	Compute, scale and shift value of y.
50	Compute x coordinate of left end of tangent segment.
60	Compute y co-ordinate of left end of tangent segment.
70	Send plotter pen to left end of tangent segment with point up.
80	Compute x co-ordinate of right end of tangent segment.
90	Compute y co-ordinate of right end of tangent segment.
100	Move pen from left end of tangent segment to right end with pen down.



PLOTTER OUTPUT OF THE PROGRAM /SINTAN/.

/BOUNCE/ is a program that simulates inelastic collision for a ball moving in a planar "slice" of a three walled court. It assumes that the ball is given an initial horizontal velocity of 100 ft./sec., and that the only force acting on the ball while in air is that of gravity.

When the ball hits a wall, the x component of velocity is reversed and decreased by 25%; when it hits the floor, the y component is reversed and decreased by 25%. The value 25% was chosen arbitarily to represent a loss in kinetic energy of motion (where does this energy go to???).

LISTING OF THE PROGRAM /BOUNCE/.

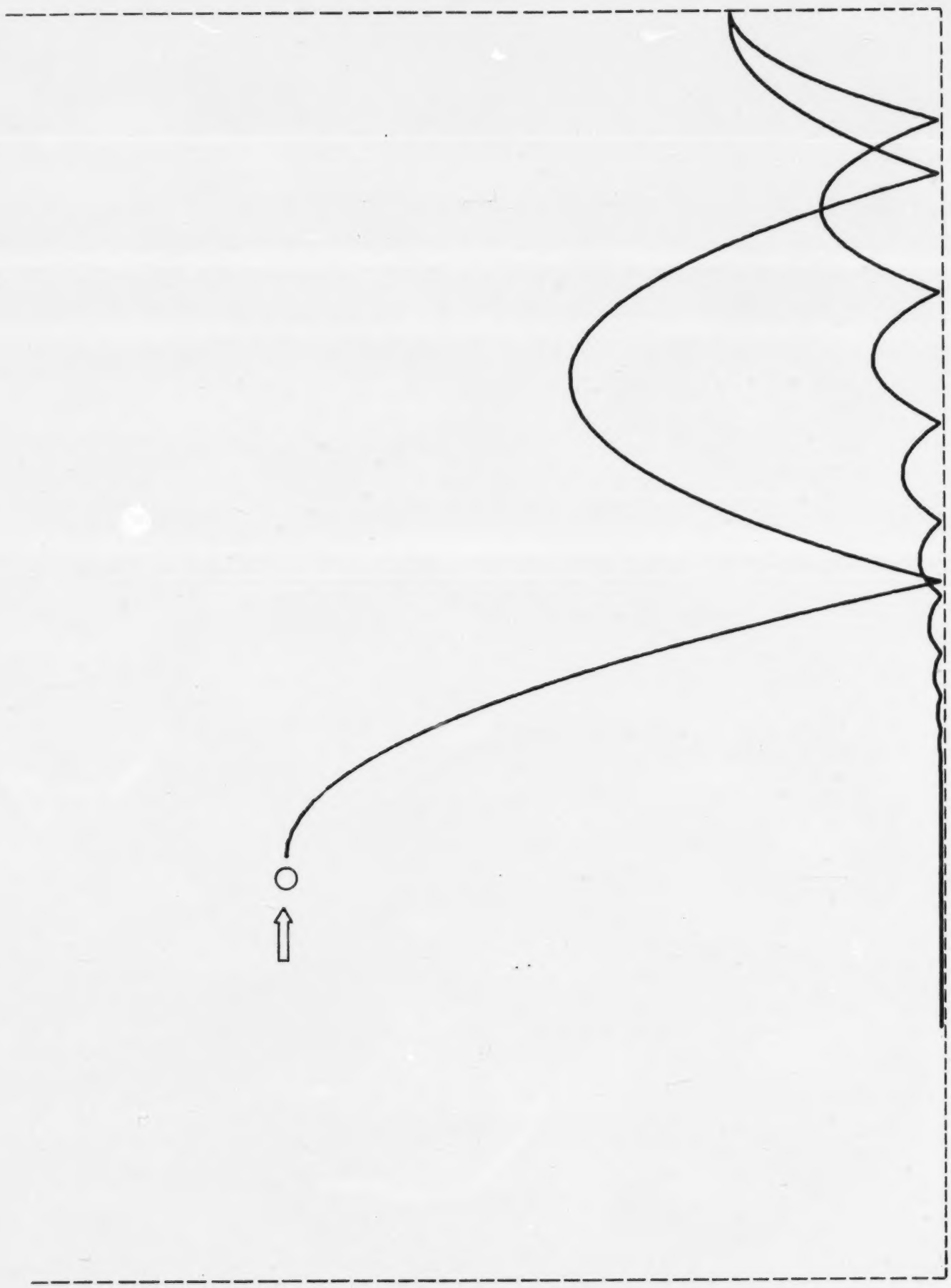
```
 FEB  8 11:55    /BØUNCE/


10 PRINT "PLTL"
20 X=3000
30 Y=7000
40 V1=100
50 V2=0
60 I=0
65 PRINT INT(X); INT(Y)
70 I=I+1
80 X=X+V1*0.04
90 IF X>1000 AND X<9000 THEN 170
91 IF X>1000 THEN 94
92 X=1000
93 GØ TØ 95
94 X=9000
95 Y=Y+V2*0.04
100 PRINT INT(X); INT(Y)
110 I=0
120 V1=(-.75)*V1
130 GØ TØ 70
170 Y=Y+V2*0.04
180 IF Y>1000 THEN 270
181 Y=1000
190 PRINT INT(X); INT(Y)
200 I=0
210 V2=(-.75)*V2
220 GØ TØ 70
270 V2=V2-1.28
280 IF I<>25 THEN 300
290 PRINT INT(X); INT(Y)
295 I=0
300 GØ TØ 70
310 END
```

EXPLANATION OF THE PROGRAM /BOUNCE/

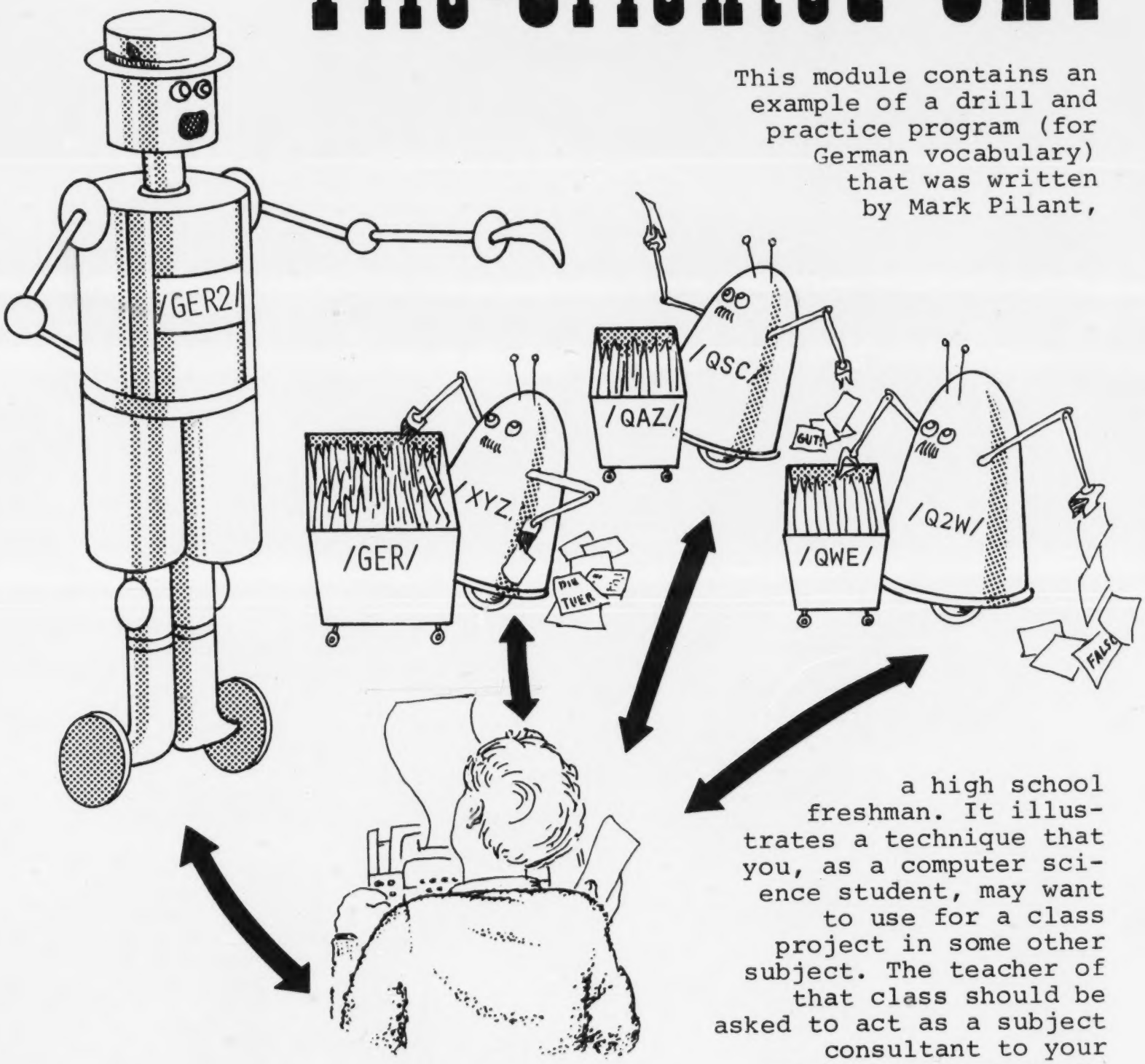
STATEMENT	
10	Starts plotting in line mode.
20-30	Set initial position of ball.
40	Sets initial horizontal (or x) component of velocity.
50	Sets initial vertical (or y) component of velocity.
60	The new position of the ball is calculated every .04 seconds; however, the position is plotted only every second. The variable I is a counter which keeps track of when the plotting should take place. Hence it is initialized to zero and when it equals 25 the position is plotted and I is set back to 0 etc.
65	The initial position is plotted.
70	I is incremented.
80	The x co-ordinate of the new position is calculated as new value = old value + horizontal velocity * change in time.
90	The "walls" are at x=1000 (left wall) and x=9000 (right wall). If the ball is between the walls, skip the bounce. That is, go to 170.
91	If the ball is not between the walls and x>1000 then x must be at the right wall. (It can't go past the wall.) So go to 94.
92	Otherwise the ball must be at the left wall. Therefore set x=1000.
94	The ball is at the right wall so set x=9000.
95	The y co-ordinate of the new position is calculated. See explanation of statement 80.

100	When the ball is at a wall, its position is plotted whether the counter is 25 or not.
110	Reinitialize counter to zero since a point was just plotted.
120	The bounce was an inelastic collision so the x velocity is decreased by 25% and reversed.
130	The case when the ball hits a wall is taken care of so compute the next position.
170	Compute the y co-ordinate of the next position (see 80).
180	If the ball is above the floor which is at y=1000 skip the bounce. That is, go to 270.
181	Otherwise, the ball is at the floor so set y=1000.
190	Plot the position when the ball is at the floor regardless of the counter.
200	Reinitialize the counter since a point was just plotted.
210	Reverse and decrease by 25% the vertical component of the velocity.
220	Compute next position.
270	In .04 seconds the force of gravity causes the vertical component of the velocity to decrease by 1.28 ft./sec.
280	See explanation of statement 60.
290	Plot the point.
295	Reinitialize counter.
300	Compute next position.



PLOTTER OUTPUT OF THE PROGRAM /BOUNCE/.

File-Oriented CAI



This module contains an example of a drill and practice program (for German vocabulary) that was written by Mark Pilant, a high school freshman. It illustrates a technique that you, as a computer science student, may want to use for a class project in some other subject. The teacher of that class should be asked to act as a subject consultant to your work as a programmer.

The picture above shows the relationship between the different parts of a file-oriented drill. The program /GER2/ controls the drill. The three files /GER/, /QAZ/, and /QWE/ (which contain vocabulary, 'congratulation' messages, and 'no-no' messages) are called on by /GER2/. They can be changed by using the 'updating' programs /XYZ/, /QSC/, and /Q2W/, respectively. When designing a file-oriented drill for another subject, you may wish to vary the arrangement described here.

SAMPLE STUDENT INTERACTION WITH /GER2/

```
GUTEN TAG.  ICH HEISSE HANS.  WIE HEISSEN SIE?
?MARK
IF YOU WANT TO TRANSLATE ENGLISH TO GERMAN TYPE ENGLISCH. IF
YOU WANT TO TRANSLATE GERMAN TO ENGLISH TYPE DEUTSCH
?ENGLISCH
WIEVIELE WOERTER WOLLEN SIE GEBEN? TIPPEN SIE DIE NUMMER
?2
TIPPEN SIE DAS DEUTSCHE WORT FUER THE RIVER
?DAS FLUSS
DAS GEHT NICHT
SIE HABEN DAS FALSCHE WORT GEGEBEN
TIPPEN SIE DAS DEUTSCHE WORT FUER THE RIVER
?DER FLUSS
WUNDERBAR
TIPPEN SIE DAS DEUTSCHE WORT FUER THE UNCLE
?DER ONKEL
GROSSARTIG
WENN SIE WEITERMACHEN WOLLEN, TIPPEN SIE MEHR :SONST NEIN
?MEHR
IF YOU WANT TO TRANSLATE ENGLISH TO GERMAN TYPE ENGLISCH. IF
YOU WANT TO TRANSLATE GERMAN TO ENGLISH TYPE DEUTSCH
?DEUTSCH
WIEVIELE WOERTER WOLLEN SIE GEBEN? TIPPEN SIE DIE NUMMER
?2
TIPPEN SIE DAS ENGLISCHE  WORT FUER DAS JAHR
?THE YEAR
FABELHAFT
TIPPEN SIE DAS ENGLISCHE  WORT FUER MUEDE
?TIREDED
FALSCH
SIE HABEN DAS FALSCHE WORT GEGEBEN
TIPPEN SIE DAS ENGLISCHE  WORT FUER MUEDE
?TIRED
SEHR GUT
WENN SIE WEITERMACHEN WOLLEN, TIPPEN SIE MEHR :SONST NEIN
?MEHR
IF YOU WANT TO TRANSLATE ENGLISH TO GERMAN TYPE ENGLISCH. IF
YOU WANT TO TRANSLATE GERMAN TO ENGLISH TYPE DEUTSCH
?ENGLISCH
WIEVIELE WOERTER WOLLEN SIE GEBEN? TIPPEN SIE DIE NUMMER
?2
TIPPEN SIE DAS DEUTSCHE WORT FUER THE DOOR
?DIE TUER
RICHTIG
TIPPEN SIE DAS DEUTSCHE WORT FUER TO SEE TO LOOK
?SEHIN
NICHT GUT GENUG
SIE HABEN DAS FALSCHE WORT GEGEBEN
TIPPEN SIE DAS DEUTSCHE WORT FUER TO SEE TO LOOK
?SEHEN
PRIMA
WENN SIE WEITERMACHEN WOLLEN, TIPPEN SIE MEHR :SONST NEIN
?MEHR
IF YOU WANT TO TRANSLATE ENGLISH TO GERMAN TYPE ENGLISCH. IF
YOU WANT TO TRANSLATE GERMAN TO ENGLISH TYPE DEUTSCH
?DEUTSCH
WIEVIELE WOERTER WOLLEN SIE GEBEN? TIPPEN SIE DIE NUMMER
?2
TIPPEN SIE DAS ENGLISCHE  WORT FUER DER QUATSCH
 ---ETC.---
```

LISTING OF THE PROGRAM /GER2/

```
10 PRINT "GUTEN TAG.  ICH HEISSE HANS.  WIE HEISSEN SIE?"
20 INPUT Q$
30  DIM E$(100),G$(100),T$(8),M$(6)
40 PR. "IF YOU WANT TO TRANSLATE ENGLISH TO GERMAN TYPE ENGLISCH. IF"
50 PR. "YOU WANT TO TRANSLATE GERMAN TO ENGLISH TYPE DEUTSCH"
60 INPUT C$
70 IF C$="ENGLISCH" THEN 100
80 IF C$="DEUTSCH" THEN 580
90 GOTO 40
100 OPEN /GER/ FOR INPUT AS 3
110 FOR X=1 TO 100
120 INPUT FROM 3:E$(X)
130 INPUT FROM 3:G$(X)
140 NEXT X
150 PR. "WIEVIELE WOERTER WOLLEN SIE GEBEN? TIPPEN SIE DIE NUMMER"
160 INPUT H
170 FOR K=1 TO H
180 LET I=NUM(100)
200 PRINT "TIPPEN SIE DAS DEUTSCHE WORT FUER ":E$(I)
210 INPUT F$
220 IF IS(F$,G$(I),0) GOTO 240
222 LET Z=0
225 LET Z=Z+1
230 GOTO 330
240 OPEN /QAZ/ FOR INPUT AS 4          (See note above.)
250 FOR G=1 TO 8
260 INPUT FROM 4:T$(G)
270 NEXT G
280 LET H=NUM(8)
290 PRINT :T$(H)
300 CLOSE 4
310 GOTO 460
330 OPEN /QWE/ FOR INPUT AS 5          (See note above.)
340 FOR Z=1 TO 6
350 INPUT FROM 5:M$(Z)
360 NEXT Z
370 LET Z1=NUM(6)
380 PRINT :M$(Z1)
390 CLOSE 5
410 PR. "SIE HABEN DAS FALSCHE WORT GESEBEN"
430  IF Z=2 THEN 450
445 GOTO 200
450 PR. "ES SOLLTE SEIN ":G$(I)
455 GOTO 540
460 NEXT K
470 IF Z>0 GOTO 490
480 PRINT "AUSGEZEICHNET"
49  PR. "WENN SIE WEITERMACHEN WOLLEN, TIPPEN SIE MEHR :SONST NEIN"
500 INPUT D$
510 CLOSE 3
520 IF D$="MEHR" THEN 30
530 GOTO 570
540 PRINT "SPRECHEN SIE MIT DER LEHRERIN"
```

NOTE: Line 100 should preferably be:
100 OPEN '150RK /GER/' FOR INPUT AS 3
where 150RK is the number where files are stored. This will allow other users to run this program.

/GER/ continued

```
550 PR. "DAS IST ALLES FUER JETZT. "
560 CLOSE 3
570 GOTO 1030
580 DIM E$(100),G$(100),T$(8),M$(6)
590 OPEN /GER/ FOR INPUT AS 3                    (See note.)
600 FOR X1=1 TO 100
610 INPUT FROM 3:E$(X1)
620 INPUT FROM 3:G$(X1)
630 NEXT X1
640 PR. "WIEVIELE WOERTER WOLLEN SIE GEBEN? TIPPEN SIE DIE NUMMER"
650 INPUT H1
660 FOR K1=1 TO H1
670 LET I1=NUM(100)
690 PRINT "TIPPEN SIE DAS ENGLISCHE  WORT FUER ":G$(I1)
700 INPUT P$
710 IF ICO(E$(I1),P$,0) GOTO 730
712 LET Z2=0
715 LET Z2=Z2+1
720 GOTO 820
730 OPEN /QAZ/ FOR INPUT AS 4                    (See note.)
740 FOR G=1 TO 8
750 INPUT FROM 4:T$(G)
760 NEXT G
770 LET H=NUM(8)
780 PRINT :T$(H)
790 CLOSE 4
800 GOTO 940
820 OPEN /QWE/ FOR INPUT AS 5                    (See note.)
830 FOR Z=1 TO 6
840 INPUT FROM 5:M$(Z)
850 NEXT Z
860 LET Z1=NUM(6)
870 PRINT :M$(Z1)
880 CLOSE 5
890 PR. "SIE HABEN DAS FALSCHE WORT GEGEBEN"
910 IF Z2=2 THEN 930
925 GOTO 690
930 PR. "ES SOLLTE SEIN ":E$(I1)
935 GOTO 1020
940 NEXT K1
950 IF Z2>0 GOTO 970
960 PRINT "AUSGEZEICHNET"
970 PR. "WENN SIE WEITERMACHEN WOLLEN, TIPPEN SIE MEHR :SONST NEIN"
980 INPUT J$
990 CLOSE 3
1000 IF J$="MEHR" THEN 30
1010 GOTO 1030
1020 PRINT "SPRECHEN SIE MIT DER LEHRERIN"
1030 PR. "DAS IST ALLES FUER JETZT. "
1040 PR. "AUF WIEDERSEHEN ":Q$
1050 END
```

FILES USED IN /GER2/

```
-COPY /QAZ/ TO TEL
RICHTIG
FABELHAFT
GUT
PRIMA
ALLERHAND
SEHR GUT
GROSSARTIG
WUNDERBAR
```

/QAZ/ is a file of positive reinforcement messages.

```
-COPY /QWE/ TO TEL
NICHT RICHTIG
FALSCH
NICHT GUT GENUG
DAS GEHT NICHT
NEIN
UNRICHTIG
```

/QWE/ is a file of negative reinforcement messages.

/Q2W/

```
5 DIM M$(6)
10 OPEN /QWE/ FOR OUTPUT AS 5
20 FOR W=1 TO 6
30 INPUT M$(W)
40 PRINT ON 5:M$(W)
50 NEXT W
60 CLOSE 5
70 END
```

/Q2W/ is a program to create or update /QWE/.

/XYZ/

```
5 DIM E$(100),G$(100)
10 OPEN /GER/ FOR OUTPUT AS 3
20 FOR X=1 TO 100
30 INPUT E$(X)
40 INPUT G$(X)
50 PRINT ON 3 :E$(X)
55 PRINT ON 3 :G$(X)
60 NEXT X
70 CLOSE 3
80 END
```

/XYZ/ is a program to create or update /GER/.

/QSC/

```
10 DIM G$(8)
20 OPEN /QAZ/ FOR OUTPUT AS 3
30 FOR X=1 TO 8
40 INPUT G$(X)
50 PRINT ON 3:G$(X)
60 NEXT X
70 CLOSE 3
80 END
```

/QSC/ is a program to create or update /QAZ/.

```
-COPY /GER/ TO TEL
THE EVENING
DER ABEND
THE FOOD THE MEAL
DAS ESSEN
THE WORK
DIE ARBEIT
THE DOCTOR
DER ARZT
THE LESSON THE TASK THE ASSIGNMENT
DIE AUFGABE
THE RAILWAY
DIE BAHN
THE MOUNTAIN
DER BERG
THE BED
DAS BETT
THE BOAT
DAS BOOT
THE BOOK
DAS BUCH
THE OFFICE
DAS BUERO
THE STATION WAGON
DER COMBI
THE THING
DAS DING
THE CORNER
DIE ECKE
THE END
DAS ENDE
THE TRIP
DIE FAHRT
THE BOTTLE
DIE FLASCHE
THE RIVER
DER FLUSS
THE FRIEND
DER FREUND
TO TRAVEL
FAHREN
 ---ETC.---
```

The file /GER/ contains the alternate English and German words on which the student is to be drilled.

This file can be changed at any time by running the program /XYZ/.

In a similar manner, the reinforcement files can be updated.

For example: /Q2W/ is used below to change the negative reinforcement file to English words--

```
-NBS
>RUN 150RK /Q2W/
?LOUSY
?ROTTEN
?STUPID ANSWER!!
?BOY ARE YOU DUMB
?NO, NO, NO, NO!!!!
?FORGET IT

-NBS
>RUN 150RK /GER2/
GUTTEN TAG. ICH HEISSE HANS. WIE HEISSEN SIE?
?FRITZ
IF YOU WANT TO TRANSLATE ENGLISH TO GERMAN
TYPE ENGLISCH. IF YOU WANT TO TRANSLATE
GERMAN TO ENGLISH TYPE DEUTSCH.
?ENGLISCH
WIEVIELE WOERTER WOLLEN SIE GEBEN?
TIPPEN SIE DIE NUMMER
?3
TIPPEN SIE DAS DEUTSCHE WORT FUER VAGRANT
?BUMMER
STUPID ANSWER!!
SIE HABEN DAS FALSCHE WORT GEGEBEN
TIPPEN SIE DAS DEUTSCHE WORT FUER HIPPIE
?SCHLAMIEL
NO, NO, NO, NO!!!!
 ---ETC.---
```

Assignment 1. Create a similar drill system for preparing students for the verbal part of college entrance examinations.

References:

1. Barron's "How to Prepare for College Entrance Examinations."
2. Barnes & Noble's "College Entrance Examinations."
3. Gale, "Building an Effective Vocabulary."

Assignment 2. Organize an experiment in your school to see if students who prepare with a computer drill program do better than a control group of students who study in other ways.